

# 44 Lecture - CS304

## Important Mcqs

### 1. What is stack unwinding?

- a) A process of releasing memory from the heap
- b) A process of releasing memory from the stack
- c) A process of clearing the call stack in response to an exception being thrown
- d) A process of allocating memory on the stack

Answer: c

### What happens during stack unwinding?

- a) All functions that were called before the exception occurred are popped off the stack
- b) All functions that were called after the exception occurred are popped off the stack
- c) All functions in the program are popped off the stack
- d) All variables in the program are popped off the stack

Answer: a

### Why is stack unwinding important?

- a) It prevents resource leaks in C++ programs
- b) It allows programs to allocate memory on the stack
- c) It improves program performance
- d) It allows programs to release memory from the heap

Answer: a

### What is the purpose of calling destructors during stack unwinding?

- a) To release any dynamically allocated memory or resources
- b) To allocate memory on the stack
- c) To improve program performance
- d) To initialize variables on the stack

Answer: a

### What happens if an exception is thrown but not caught?

- a) The program terminates immediately
- b) The program continues to execute normally
- c) The program enters an infinite loop
- d) The program enters a state of undefined behavior

Answer: a

### What is the role of the try block in stack unwinding?

- a) It contains the code that may throw an exception
- b) It contains the code that is executed if an exception is caught
- c) It contains the code that is executed if an exception is not caught
- d) It contains the code that is executed before stack unwinding begins

Answer: a

### What is the role of the catch block in stack unwinding?

- a) It catches and handles exceptions

- b) It releases memory from the heap
- c) It initializes variables on the stack
- d) It allocates memory on the stack

**Answer: a**

**What happens if a function throws an exception but does not have a catch block?**

- a) The program terminates immediately
- b) The program continues to execute normally
- c) The exception is caught by a catch block in a higher function on the call stack
- d) The program enters a state of undefined behavior

**Answer: c**

**When are destructors called during stack unwinding?**

- a) In reverse order of construction
- b) In the order of construction
- c) Randomly
- d) It depends on the implementation

**Answer: a**

**What is the difference between stack unwinding and stack overflow?**

- a) Stack unwinding is intentional, while stack overflow is unintentional
- b) Stack unwinding releases memory from the stack, while stack overflow overwrites memory on the stack
- c) Stack unwinding occurs during exception handling, while stack overflow occurs when the stack becomes too full
- d) Stack unwinding only occurs in C++, while stack overflow can occur in any programming language

**Answer: c**